CLAIMS

1. A method of providing data entities of a database with a time dependent value for an attribute of the entity, the method comprising:

   i) having values of attributes of said entities in said database;

   ii) for said entities providing a valid time start time associated with each said value of said attribute;

   iii) for said entities providing a valid time end time associated with each said value of said attributes;

   iv) for said entities providing a transaction time associated with said valid time start time and said valid time end time for each said value; whereby

   v) when querying said database an appropriate said transaction time is used so as to ensure that an appropriate value for said attribute is used and that appropriate valid time start and end times are used.

2. A method according to claim 1 wherein said valid end time is provided by either: (i) storing said valid end time entered in said database associated with said value for said attribute; or (ii) determining said valid end time from a valid start time of another value of said attribute stored in said database.

3. A method according to claim 1 wherein changes to said values are achieved by inserting new entries for linked transaction time, valid time start time, valid time end time, and associated attribute value over a thereby specified valid time period, there being no actual delete operator used to change said values.

4. A method according to claim 3 wherein there is no update operator, update being achieved by insertion.

5. A method according to claim 3 wherein a logical delete operation for a value of an attribute of an entity is achieved by an insert operation which sets the said value to Non Existent for a period in which it is intended to delete said attribute or value.

6. A method according to claim 1 comprising providing said values for data entries in a relational database.

7. A method of obtaining information from a database comprising providing entities in said database with time dependent values for attributes in accordance with claim 1 and querying said database for a value for selected data entity(s).

8. A method according to claim 7 comprising having Temporal Data Dictionary having a set of objects associating specific entities in said database with specific functionality-related parameters which are used in controlling how a processor handles a query relating to said entities, said Temporal Data Dictionary maintaining data integrity.

9. A method according to claim 8 comprising having a relational database, and wherein the Temporal Data Dictionary also maintains referential integrity rules for data entities that are referentially related, and wherein referential integrity for data entries during a response to queries of the database is maintained by the temporal data dictionary, rather than by specific application - level routines running on a database server processor.

10. A method according to claim 1 wherein data relating to each specific data entity is held in corresponding data-entity associated Time Cubes that

comprise valid and transaction time related values for different attributes in different attribute-slices of said Time Cube.

11. A method according to claim 10 wherein for each Time Cube there is a single primary key attribute having associated values over transaction and valid time, and wherein in the Time Cubes there is another, or a plurality of other foreign key attributes having a value, or values, which are valid and/or transaction time dependent.

12. A database having a model of entities on said database in which each entity has an associated Time Cube:

i) said Time Cubes comprising at least one attribute having allowable values in X-Time and an associated Y time, X time being the whole system time from start of system time to end of system time within which valid time will exist for said entity, and Y-Time being system – generated time representative of transaction time at which specific allowable values are true.

13. A database according to claim 12 wherein said model allows insert only events, with update and delete functions being achievable by insert events.

14. A method of providing database software comprising taking an existing database core code which has update and/or delete functions and replacing said update and/or delete functions with an insert-only core code model so as to provide different values of attributes associated with entities on the database, said different values being associated with respective transaction times at which they are true, said database being interrogatable in a manner to query what said values are at different transaction times, and wherein the insert-only core code provides a user-experienced pseudo update and/or delete function.

15. A method of holding data in a database comprising using a data model to hold data concurrently and consistently represented over time, the method comprising including time as a characteristic associated with values of attributes of entities on said database, and achieving update and delete of values of attributes of entities by using inserts, adding to the database, to achieve logical update and logical delete, without any requirement for actual code-level update and actual code-level delete operations, and wherein logical delete update and insert are achieved by physical insert.

16. A method of modifying the value of an entity in a database comprising for each entity having one or more attributes, each attribute having one or more value over valid time, and storing for each value a start time of valid time from which the value applies, an end time of valid time to which the value applies, and a transaction time at which said start and end times in valid time apply.

17. A method of modifying the value of an entry in a database comprising for the entry having one or more attributes, each attribute having one or more values over valid time, and storing for each value a start of valid time from which the value applies, and a transaction time at which the start of valid time applies.

18. A method of modelling changes in values of attributes in time in a database comprising for each attribute of an entity in the database having summary positions associated with corresponding transaction times, each summary position, specific to a transaction time, having for each attribute a value, a start time in valid time at which said value applies, and optionally an end time in valid time at which said value applies; and wherein said transaction time represents a time at which associated values, and start times, and optionally end times for those values, are taken to be true; and wherein an additional summary position is generated whenever an insert,

logical update, or logical operation is performed on the database, and wherein logical delete and logical update are performed by inserting a new summary position with appropriate transaction time and start and optionally end valid times for an associated value for an attribute.

19. A method of providing data entities of a database with a time dependent value for an attribute of the entity, the method comprising:

i) having values of attributes of said entities in said database;

ii) for said entities providing a valid time start time associated with each said value of said attribute;

iii) for said entities providing a transaction time associated with said valid time start time for each said value; whereby

iv) when querying said database an appropriate said transaction time is used so as to ensure that an appropriate value for said attribute is used.

20. In a temporal database system having a temporal database storing a database table comprising a plurality of data rows, said rows relating to attributes of a database entity and storing values for said attributes organised into particular respective database fields, an improved method for granting access during the modification of the information in a database fields comprising locking the field being accessed for modification and not locking other fields in the row that contains the field being accessed for modification, this being achieved by associating a transaction time with the information in each field, said transaction time being related to the system time at which the information was inserted into the database, permitting insert only operations on the database, with alter and/or delete functionality performed by insert-only addition of another value of an attribute held in a certain field in an additional field differentiated from other fields for the same attribute by its associated transaction time, and selecting an appropriate field for access by selecting an appropriate transaction time, as well as selecting said attribute to be queried or modified.